PREPRINT UCRL- 81990

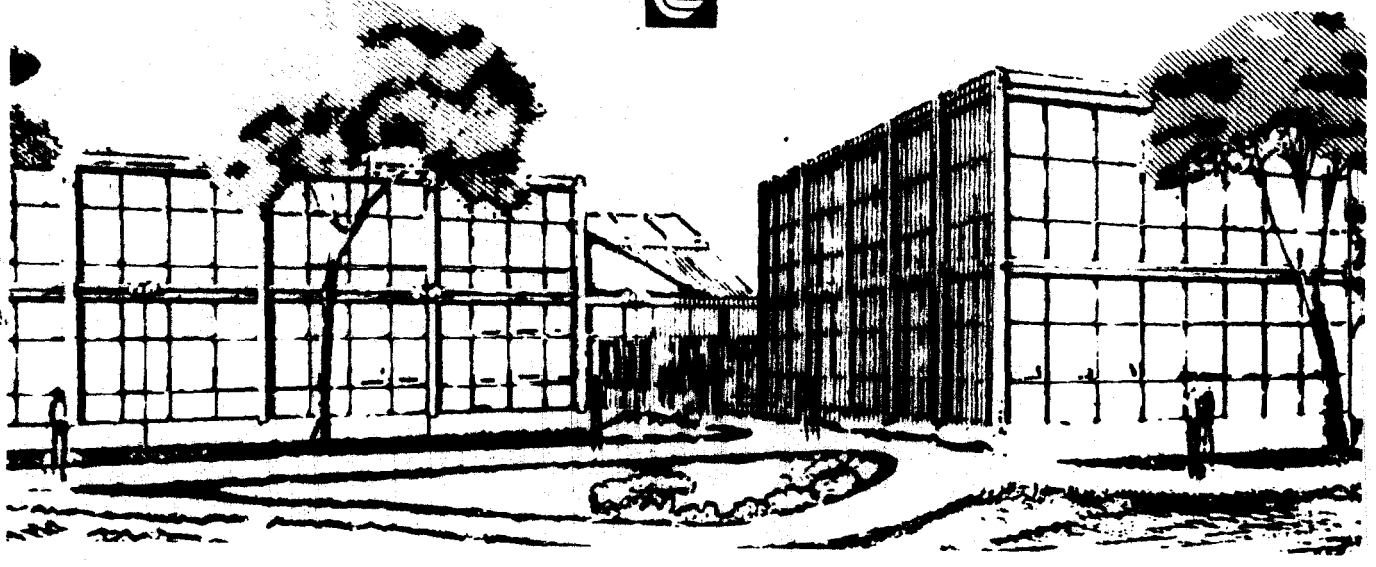
Lawrence Livermore Laboratory

TITLE: DATA FLOW AND HIGH SPEED COMPUTATION

AUTHOR: TIMOTHY E. RUDY

DATE: DECEMBER 5, 1978

THIS PAPER WAS PREPARED FOR SUBMISSION TO:
CERT - Centre d'etudes et de recherches de Toulouse, France , February 1979



ABSTRACT

The University of California's Lawrence Livermore Laboratory and MIT's Laboratory of Computer Science are evaluating data driven architectures for high speed computation. This paper describes the status of research at LLL and estimates the thruput requirements for a packet communications architecture to achieve a gigaflop for LLL application programs.

"Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore Laboratory under contract number W-7405-ENG-48."

Since its inception, the University of California's Lawrence Livermore Laboratory (LLL) has effectively utilized the most powerful computing machines available to facilitate the applied energy research programs at LLL. The Livermore Computer Center consists of four Control Data Corporation (CDC) 7600's and two CDC STAR-100's which communicate with mass storage, graphic and high speed printing devices through a loosely coupled point to point network designed and constructed at LLL called OCTOPUS. There are approximately 1200 terminals connected to the network.

Our interest in Data Flow architectures is based on increasing our computational capabilities by approximately two orders of magnitude over existing "conventional" alternatives. Since the physics models at LLL which consume the majority of the available computing resources are typically limited by floating point thruput, we measure performance in units of millions of floating point operations per second (MFLOPS). For a reference point a CDC 7600 is 3-6 MFLOPS. With a high degree of vectorization (greater than ninety percent) application programs on the CDC STAR-100 have attained approximately 15 MFLOPS. We will acquire a CRAY-1 in early 1979 and anticipate performance in the 6-30 MFLOPS range.

Since a viable Data Flow machine would probably not be available until the late 1980's, a desirable performance level should be in excess of 1000 MFLOPS. This projection is hopefully conservative since a study funded by the National Aeronautics and Space Administration (NASA) for a 1000 MFLOPS capability by 1985 for a limited class of applications has resulted in proposed architectures by the Burroughs Corporation and CDC which exceed the requirements (1).

Application Programs

A broad spectrum of computational models have been developed to support the energy research programs at LLL. These simulation codes for fluid dynamics, heat flow, laser fusion, plasma simulation, etc. use a variety of numerical techniques (finite difference, Monte Carlo, particle in cell).

A typical application program consists of 10000 lines of FORTRAN of which approximately 2000 lines dominate the execution time. However, the application program is only one component of a code system of 200000 lines of FORTRAN which consists of input and output processing programs.

A two dimensional fluid dynamics code, depending upon the formulation and additional physics, may require 10000 - 100000 zones with 10-100 global variables per zone. Since random access memory is an expensive component and limited resource in high-speed computers, the user, in problem definition, and programmer, in implementation, maximize the utilization of memory. This implies a gross violation of the single assignment rule and the explicit dependence on side effects to increase efficiency.

1000 MFLOPS and Data Flow

The following measurements were made on a CRAY-1 for a two- dimensional heat conduction program which uses an Alternating Direction Implicit (ADI) method to solve the diffusion equation. Although written in FORTRAN, the program is completely vectorized. There are 11 global variables associated with each zone. This is the optimal number of variables to minimize the number of floating point operations. Each operation ($\sqrt{\ }$, /, $\pm$, *, AMAX) is considered as one floating point operation (FLOP).

ADI

Grid Size	MFLOPS	FLOP	References	NSEC / FLOP	NSEC / Memory Reference
16 x 16	20.8	13600	12667	48.1	51.7
64 x 64	32.8	250528	231464	30.5	33.0
128 x 128	33.7	1025184	946075	29.7	32.1

For a Data Flow architecture which is based on packet communications (2) we can extrapolate the required thruput rates for the routing network, packet memory system and scalar processor.

To achieve a 1000 MFLOPS rate the following estimates are made:

Routing Network	.5 nsec/packet
Memory System	1.1 nsec/access
Scalar Processor	1 nsec/operation

With an estimated 20 microsecond latency for the routing network (instruction enable to delivery of results to other instruction cells) (3), 20000 instructions must be in the routing network throughout the calculation to avoid latency delays.

If we ignore instruction cells needed for bookkeeping operations, the 128 x 128 ADI problem would require $\sim 2 \times 10^6$ instruction cells. The instruction cell memory requirement could be reduced by using re-entrant procedures.

Another example is an increasingly popular method for solving systems of linear equations used by the major laser fusion code at LLL, the Incomplete Cholesky-Conjugant Gradient Method (ICCG) (4). A FORTRAN version of the ICCG algorithm implemented on a CRAY-1 produced the following performance measurements while solving a 51 x 51 matrix:

18.4 MFLOPS
54.3 nsec/FLOP
40.5 nsec/Memory Access

Repeating the extrapolation to a 1000 MFLOPS Data Flow machine for ICCG:

Memory System	.75 nsec/access
Routing Network	.42 nsec/packet

A more optimal implementation of the ICCG algorithm (25% of the floating point operations are performed in scalar mode) would decrease the thruput requirements.

ADI and ICCG algorithms represent 5-20% of the execution time of an application code. Both methods use recurrence equations which may introduce significant delays in the routing network. The degree to which these data dependent operations can be overlapped by unrelated operations is unknown.

The last example of computational requirements for LLL application codes is based on an instruction stream analysis for one time step of the most heavily used 2-D code at Livermore (5):

5000 instructions per zone
  1600 floating point operations
  1700 memory references
333 microseconds compute time per zone

Our mythical gigaflop machine would require a routing network which process packets at .32 nanoseconds and a memory system which can process requests in .94 nanoseconds.

Assuming the hardware allows sufficient concurrency to meet the bandwidth requirements, achieving a 1000 MFLOPS processing capability is dependent on the amount of parallelism in a Data Flow program. A comprehensive analysis is impractical since neither a high level language or simulator is available. It is possible, however, to estimate the fraction of instructions which must be "active" to prevent latency. For a 100 x 100 zone problem there would be 33×10^6 scalar processor and 17×10^6 memory system instructions for a given time step. If 20×10^3 instructions are required to saturate each pipeline, the .06% of the scalar and .12% of the memory instructions would saturate the machine.

Data Flow Research and LLL

For about the last two years, LLL and the Computations Structures Group at the MIT Laboratory for Computer Science have collaborated in Data Flow research which has focused on high speed computing possibilities for LLL applications. Two Data Flow workshops have served to increase communication in the Data Flow research community, enlightened LLL staff members to a variety of research activities in the U.S. and Europe and provided a forum for LLL to quantify present and future computing requirements.

LLL's role in this collaborative effort with MIT has focused on evaluating architectural proposals and participating in Data Flow language design. To assist researchers at MIT and other universities, we have written and documented a two dimensional Lagrangian inviscid compressible hydrodynamics program with heat conduction and a tabular ideal gas equation of state (5). Additional sample programs will be developed in the near future.

LLL and MIT staff members are currently evaluating a proposed high level programming language for data driven computation designed by MIT and LLL personnel. VAL (a Value-Oriented Algorithmic Language) is a side effect free applicative language consisting of function modules.

Future activities at LLL include the translation of programs into VAL, development of a parser and syntax checker, provision of a tutorial language manual with examples to assist programmers in rewriting application codes in VAL and development of a simulation capability.

References

(1) _____ , "Future Computer Requirements for Computational Aerodynamics", NASA CP-3032, Oct., 1977.

(2) Dennis, J.B., "Packet Communication Architecture", Proceedings of the 1975 Sagamore Computer Conference on Parallel Processing, August, 1975, 224-229.

(3) Dennis, J.B. and K.S. Weng, "Application of Data Flow Computation to the Weather Problem," High Speed Computer and Algorithm Organization, (D.J. Kuck, D.H. Lawrie, and A.H. Sameh, Eds.), 1977, 143-157.

(4) Kershaw, David S., "The Incomplete Cholesky-Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations", Journal of Computational Physics, Jan., 1978, 43-65

(5) C.P. Hendrickson, private communication

(6) Crowley, W.P. and C.P. Hendrickson, T.E. Rudy, "The Simple Code," Lawrence Livermore Laboratory publication, UCID 17715, 1978.